UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/057,002	01/24/2002	Shlomo Wygodny	MUTEK.4C1CP1	5496

20995 7590 03/01/2005

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
MITCHELL, JASON D	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 03/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/057,002	WYGODNY ET AL.	
	Examiner	Art Unit	
	Jason Mitchell	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 January 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _____ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This application claims priority to U.S. patent application 09/923,972 filed on 08/07/2001 which is a continuation of 09/748,752 filed on 12/26/2000, which claims priority to U.S. provisional application 60/172,101 filed on 12/32/1999.

2. Claims 1-36 are pending in this case.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. **Claim 3 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement.** The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The claim recites 'a trigger is activated every predefined interval of time', indicating that only a specific trigger is activated at the predefined interval of time. No disclosure was found in the specification for this limitation. What was found in the specification is support for a limitation where tracing on the whole starts or stops at a specific time (i.e. par [0017]).

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. **Claim 8 recites the limitations "said display screen" in line 16, and "said developer" in line 17. There is insufficient antecedent basis for these limitations in the claim.** Examiner's best understanding will be used in the treatment of this claim and "said display screen" will be assumed to refer to "a display screen", and "said developer" will be assumed to refer to "a developer".

7. **Claim 23 recites the limitation "said condition" in 11. There is insufficient antecedent basis for this limitation in the claim.** Examiner's best understanding will be used in the treatment of this claim and "said condition" will be assumed to refer to "a condition of said trigger".

8. **Claims 27 and 31 recite the limitation "said trace control dataset" in lines 3 and 15 respectively. There is insufficient antecedent basis for this limitation in the claims.** Examiner's best understanding will be used in the treatment of this claim and "said trace control dataset" will be assumed to refer to "said trace data".

9. **Claim 30 recite the limitation "said trace information" in line 10. There is insufficient antecedent basis for this limitation in the claims.** Examiner's best understanding will be used in the treatment of this claim and "said trace information" will be assumed to refer to "said trace data".

Double Patenting

10. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA

1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

11. Claims 6-35 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-32 of U.S. Patent No. 6,202,199 (199) in view of US 5,732,210 to Buzbee (Buzbee).

Regarding Claim 6: Claim 2 of the 199 reference discloses the limitations in the instant claim except the limitation of the trace control information including one or more pairs of triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 7: Claim 1 of the 199 reference discloses the limitations in the instant claim except the limitation of the trace control information including one or more pairs of triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 8: Claim 5 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the

debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 9: Claim 6 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 10: Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 11: Claim 12 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would

have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 12: Claim 7 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 13: Claim 13 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is

satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 14: Claim 14 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a

breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 15: Claim 15 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 16 Claim 16 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is

satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 17: Claim 18 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a

breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 18: Claim 19 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 19: Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification

of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 20: Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers and an occurrence of an event corresponding to execution of a source line.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts

execution ... or performs some specified action') and that said triggers are in response to the execution of a source line (col. 2, lines 6-8 'when it reaches a certain code location').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 21: Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers and that said action includes writing a comment to said trace long.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') wherein said action includes writing a comment to said trace log (col. 2, lines 1-2 'the debugger is then instructed to print the value of "i"').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 22: Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers and said triggers including at least on condition, said condition specifying whether said trigger causes the execution of said at least one action.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') and that said triggers include at least one condition, said condition specifying whether said trigger causes the execution of said at least one action (col. 2, lines 8-10 'If the condition is satisfied, the debugger ... performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 23: Claim 5 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers and said condition comprises a logical expression based on a variable.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') said event comprises a logical expression based on a variable (col. 2, lines 14-16 'if "i" equals 3').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a

breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 24: Claim 5 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers and said condition comprises a logical expression based on a variable and said trace log information is based on conditional trace control information.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') said event comprises a logical expression based on a variable (col. 2, lines 14-16 'if "i" equals 3') and generating a log (col. 2, lines 1-2 'debugger is then instructed to print the value of "i"') based on conditional trace control information (col. 2, lines 8-10 'If the condition is satisfied').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

Regarding Claim 25: Claim 22 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 26: Claim 23 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 27: Claim 26 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been

motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 28: Claim 24 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 29: Claim 25 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 30: Claim 28 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by

Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 31: Claim 29 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 32: Claim 30 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 33: Claim 31 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is

satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 34: Claim 32 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ...

performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 35: Claim 33 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Regarding Claim 36: Claim 33 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking

actions in response to triggers and said trace control information comprising triggers and actions and a trace tree showing the results of actions taken in response to triggers. Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

Claim Rejections - 35 USC § 102

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

13. Claims 1-2 and 5 are rejected under 35 U.S.C. 102(b) as being anticipated by US 5,265,254 to Blasciak (Blasciak).

Regarding Claim 1: Blasciak discloses a method of tracing the execution of a computer program comprising: generating trace control information which specifies triggers and associated actions (col. 7, lines 33-37 'the code marked by the software engineer'); tracing executing of the computer program according to the trace control information, such that when one of said triggers occurs the corresponding action is performed (col. 3, lines 29-34 'write on procedure entry and exit'); and generating a trace log of said tracing, wherein the trace log reflects said actions performed during tracing (col. 8, lines 28-34 'a time-stamped trace list').

Regarding Claim 2: The rejection of claim 1 is incorporated; further Blasciak discloses said triggers include at least the execution of the assembly code generated from a function entry or exit (col. 3, lines 29-34 'write on procedure entry and exit').

Regarding Claim 5: the rejection of claim 1 is incorporated; further Blasciak discloses said actions include at least writing to said trace log the stack dump of functions active at a time of a trigger (col. 3, lines 29-34 'task table dumps').

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over **US 5,265,254 to Blasciak (Blasciak) in view of 5,732,210 to Buzbee (Buzbee)**.

Regarding Claim 4: The rejection of claim 1 is incorporated; further Blasciak discloses at least some of said actions are activated only if a plurality of conditions are satisfied while said triggers are activated (col. 8, lines 36-38 'Other parameter contents may also be monitored'), but does not disclose that said conditions including comparing values of data passed in said execution to other values, however Blasciak does disclose the use of conventional breakpoint technology (col. 8, lines 36-41 'such as break points ... Implementation of such measurements are believed to be well within the skill of those skilled in the art').

Buzbee teaches that triggers ('conditional breakpoints') may include comparing values of data passed in said execution to other values (col. 2, lines 14-16 'if "i" equals 3') in an analogous art for the purpose of providing a debugger the ability to ignore triggers if the state of execution is not of interest (col. 2, lines 10-11 'If the condition is not satisfied, the debugger allows the program to continue').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use triggers ('conditional breakpoints') as taught by Buzbee to provide the code markers of Blasciak the ability to ignore triggers if the state of execution is not of interest (col. 2, lines 10-11 'If the condition is not satisfied, the debugger allows the program to continue')

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. US 5,421,009 to Platt; 5,446,876 to Levine et al.; 5,483,468 to

Chen et al.; 5,513,317 to Borchardt et al.; 5,848,274 to Hamby et al.; US 5,870,606 to Lindsey; 6,263,456 to Boxall et al.; and US 6,467,052 to Kaler et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Mitchell
2/16/05

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100